

Linguissime

Spécifications

Version 1.0

4/04/2016

Historique des révisions

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Auteur** |
| **4/04/2016** | 1.0 | Première version | Guillaume Danguin |
| 6/04/2016 | 2.0 | Deuxième version | Grandiere Antoine |
|  |  |  |  |
|  |  |  |  |

Sommaire

1 Introduction [6](#__RefHeading___Toc254870962)

1.1 Contexte initiale, historique et vision [6](#__RefHeading___Toc254870963)

1.2 Mission [6](#__RefHeading___Toc254870964)

1.3 Objectifs [6](#__RefHeading___Toc254870965)

1.4 Glossaire [6](#__RefHeading___Toc254870966)

1.5 Documents de référence [6](#__RefHeading___Toc254870967)

2 Description générale [7](#__RefHeading___Toc254870968)

2.1 Acteurs [7](#__RefHeading___Toc254870969)

2.2 Cas d’utilisations [7](#__RefHeading___Toc254870970)

3 Spécifications fonctionnelles [9](#__RefHeading___Toc254870971)

3.1 Carte de navigation [9](#__RefHeading___Toc254870972)

3.2 Détails des cas d’utilisations [9](#__RefHeading___Toc254870973)

4 Spécifications non fonctionnelles [13](#__RefHeading___Toc254870974)

4.1 Environnement opérationnel [13](#__RefHeading___Toc254870975)

4.2 Contraintes de conception et d’implémentation [13](#__RefHeading___Toc254870976)

4.3 Documentation utilisateur [13](#__RefHeading___Toc254870977)

4.4 Exigences de performance [14](#__RefHeading___Toc254870978)

4.5 Exigences de sécurité [14](#__RefHeading___Toc254870979)

4.6 Attributs de qualité [14](#__RefHeading___Toc254870980)

# Introduction

## Contexte initial, historique et vision

Ayant fait le constat qu'il n'existe que très peu d'applications linguistiques gratuite et de très bonne qualité, nous souhaitons créer une plateforme d'apprentissage de langues centralisé autour d'exercices interactif.

## Mission

Linguissime est une application web accessible sur mobile, tablette et ordinateur permettant de créer facilement des exercices interactif en anglais et de les réviser.

## Objectifs

* Créer une fonctionnalité permettant de créer des exercices interactifs en moins de 5 minutes.
* Implémenter un moteur de recherche fournissant des résultats en dessous de 10 ms.
* Créer une fonctionnalité de gestion de d'images permettant de gérer au minimum 10 000 images.
* Mettre en production notre application sur une instance EC2 Amazon Web Service et faire du versionning de notre application.
* Créer notre propre framework CSS en ayant au moins 20 composants réutilisable nous permettant de concevoir le visuel de notre application et de ne pas dépendre des frameworks CSS open-source.
* Créer une fonctionnalité permettant d'envoyer des emails automatiques.
* Créer un système permettant de gagner des points et niveau pour un utilisateur.
* Créer un système de parrainage ou un utilisateur peut envoyer une invitation à un de ses contact en spécifiant l'email de son contact.
* Créer une un système permettant de créer des exercices interactifs à travers une configuration par défaut.
* Créer une fonctionnalité permettant à un utilisateur de visualiser son historique de révision.
* Créer un panel administrateur permettant de visualiser le nombre d'utilisateurs, d'exercices crée et de supprimer des exercices ou utilisateurs.
* Mis en place d'une stratégie SEO afin d'optimiser notre référencement.

## Glossaire

### Termes du domaine

### Termes techniques

* Instance : Serveur virtuel
* ORM : Couche d'abstraction de la base de données

## Documents de référence

<http://www.php-fig.org/psr/psr-1/>

<http://www.php-fig.org/psr/psr-2/>

[http://www.php-fig.org/psr/psr-3/](http://www.php-fig.org/psr/psr-2/)

http://www.php-fig.org/psr/psr-4/

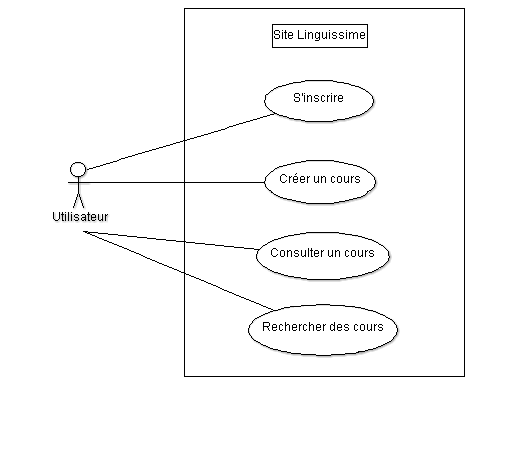
# Description générale

## Acteurs

|  |  |
| --- | --- |
| **Nom** | **Description** |
| Guillaume Danguin, Antoine Grandiere | Créateurs et développeurs du projet |
| Olivier Spinelli, Antoine Raquillet | Enseignants suiveurs |

## Cas d’utilisations

### Diagramme de cas d’utilisation



### Listes des cas d’utilisations

|  |  |  |  |
| --- | --- | --- | --- |
| **Nom du cas d’utilisation** | **Acteur(s)** | **Priorité** | **Risques** |
| Créer un exercice | Utilisateur | Haute | Haut |
| Réviser un exercice / Gagner des points et niveaux | Utilisateur | Haute | Haut |
| Recherche un exercice existant | Utilisateur | Haute | Moyen |
| Inviter un ami | Utilisateur | Moyenne | Moyen |
| Configurer son compte | Utilisateur | Moyenne | Moyen |
| Accès à une aide en ligne (faq) | Utilisateur | Faible | Faible |

# Spécifications fonctionnelles

## Carte de navigation

L'objectif de la carte de navigation est d'exprimer les chemins d'interface principaux dans le système. Ces chemins sont les chemins principaux à l'écran et ne reprennent pas la totalité des possibilités. Elle peut être considérée comme une carte routière de l'interface utilisateur.

Un diagramme de navigation peut être représenté grâce à un diagramme d’activité

## Détails des cas d’utilisations

S'inscrire : L'utilisateur peut créer un compte via notre application.

Connexion : L'utilisateur peut se connecter à l'application une fois inscrit.

Créer un exercice : L'utilisateur peut créer un exercice simplement.

Réviser un exercice : L'utilisateur peut réviser un exercice existant.

Recherche un exercice : L'utilisateur peut recherche les exercices déjà créer.

Inviter un ami : L'utilisateur peut envoyer une invitation à un ami en spécifiant l'adresse email de son ami.

Consulter des profils : L'utilisateur peut visualiser les profils des membres de l'application, y compris son propre profil.

Configurer son compte : L'utilisateur peut modifier ses informations, changer son mot de passe, ajouter ou modifier son image de profil.

Consulter ses exercices : L'utilisateur peut voir les exercices qu'il a crée .

Consulter ses statistiques : L'utilisateur peut vérifier son niveau, ses badges acquis, points et ses scores.

#### Acteur principal :

L'acteur principal est un étudiant.

#### Préconditions :

L'acteur est authentifié.

#### Scénario nominal :

L'acteur (étudiant) arrive sur la page d'accueil, il clique sur le seul et unique bouton de la page qui lui affiche un tutoriel interactif simplifié ( tutoriel) afin de montrer l'interet et les possibilités de l'application. A la fin du tutoriel, une page d'inscription.

#### Extensions:

L'acteur (étudiant) arrive sur la page d'accueil et il clique sur le lien connexion qui affiche la page connexion.

L'acteur (étudiant) arrive sur la page d'accueil et il clique sur le lien connexion qui affiche la page inscription.

#### Interface utilisateur:

Représenter les écrans de l’application. Il convient de rappeler que la maquette n’est pas le produit final !

Vous pouvez pour cela dessiner les écrans à la main ou utiliser un outil spécifique de création d’interface utilisateur (exemple : <http://www.mockupscreens.com/>, http://c2.com/cgi/wiki?GuiPrototypingTools)

# Spécifications non fonctionnelles

## Environnement opérationnel

L'application fonctionnera sous le framework Symfony avec une base de données Mysql. Notre application devra communiquer réguliérement avec Algolia pour la partie moteur de recherche.

## Contraintes de conception et d’implémentation

- Site web en responsive design ( disponible sur smartphone, tablette et ordinateur)

- Mise en production sur l'infrastructure d'amazon web services

- Implémentation de composants css orienté objet

## Documentation utilisateur

Il y aura un bouton sur la page d'accueil permettant de faire un exercice interactif « test » afin de montrer l'interet de l'application à l'utilisateur.

## Attributs de qualité

Les attributs de qualité importants pour les utilisateurs sont :

* Disponibilité : L'application sera disponible 24h/24h
* Interopérabilité : Echange de données avec Algolia ( moteur de recherche )
* Fiabilité : Pas d'erreur 404
* Robustesse : Le moteur de recherche de l'application sera tolérant aux fautes.
* Utilisabilité : L'application sera simple d'utilisation
* Sécurité : L'application protégera les comptes des utilisateurs et leurs données, notamment en cryptant les mots de passes des utilisateurs .

Les attributs de qualité importants pour les développeurs sont :

* Maintenabilité : Il sera simple de faire évoluer l'application notamment avec du versionning.
* Testabilité : Des test unitaires seront implémentés